# Psycholinguistics: Sentence Processing II

9.59; 24.905

February 17, 2005

Ted Gibson

# Today's lecture

1. Parsing
    1. Top-down
    2. Bottom-up
    3. Left-corner
    4. (Chart parsing: see reading.)

2. Human sentence comprehension: How to address the question of how sentences are comprehended.

3. Information sources used in sentence comprehension.

1. What is parsing?
2. Parsing strategies
   1. Top-down
   2. Bottom-up
   3. Left-corner
   4. (Chart parsing)

# What is parsing?

(1) The dog bites the man.

(2) The man bites the dog.

(3) *The dog bites man the.

(1) = boring, (2) = interesting,
(3) = not English

# What is parsing?

- (1), (2), and (3) have the same words
- BUT: structure different ➔ different meaning for (1) and (2)
- Not every sentence structure is possible: (3)
- A **<u>grammar</u>** tells you what are possible sentence structures in a language:
  - S $\rightarrow$ NP VP
  - NP $\rightarrow$ Det N
  - etc.

# Why is parsing hard?

- Infinite number of possible sentences
  - We can understand and say sentences we never heard before.
  - Therefore representations of sentences' meanings cannot be stored in and retrieved from memory.
- Ambiguity
  - *The man saw the woman on the hill with the telescope*.
  - Word-level ambiguity: *saw*
  - Phrase-level ambiguity: PP-attachments

# What is parsing?

Parsing:

discover how the words in a sentence can combine, using the rules in a grammar.

# What is parsing?

- Parser
  sentence → representation of meaning
- Generator
  representation of meaning → sentence

# Parsing strategies - intro

- Our sentence: *The man likes the woman*
- Our grammar:

| | | |
|---|---|---|
| S | → | NP VP |
| NP | → | Det Noun |
| VP | → | Verb NP |
| | | |
| Det | → | the |
| Noun | → | man |
| Noun | → | woman |
| Verb | → | likes |
| Verb | → | meets |

# Parsing strategies - intro

- Our grammar is unambiguous: not the case in real life
  - VP → Verb *I walk*
  - VP → Verb NP *I eat the apple*
  - VP → VP PP *I see you with the telescope*

# Parsing strategies - intro

- How to deal with ambiguity in grammar:
  - Serial approach:
    - try one rule at a time, then backtrack if necessary
    - need to specify which rule to start with
  - Parallel approach:
    - work on all alternative rules
    - need data structure that can contain set of parse trees

# Parsing strategies - intro

- Top-down parsing:
  - Start by looking at the rules in grammar
  - See if the input is compatible with the rules
- Bottom-up parsing:
  - Start by looking at input
  - See which rules in grammar apply to input
- Combination of top-down and bottom-up:
  - Only look at rules that are compatible with input

# Top-down - intro

- Assume that the input will eventually form a sentence
- Invoke S-node and all its possible extensions
- Keep expanding nodes until you find matching input
- Stack: keep track of what you still need to find in order to get grammatical sentence

# Top-down - intro

- Ambiguity terminology (not relevant to our unambiguous grammar):

  If a LHS has more than one RHS:

  - **non-deterministic** = parser does not specify expansion order
  - **deterministic** = parser specifies expansion order

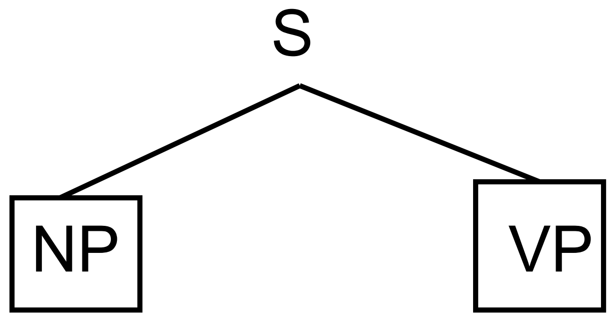
# Parsing - intro

- Notation: Elements with boxes around them need to be stored (e.g., on a **stack**) from one parse state to the next as separate items:
    - Partially processed phrase structure rules
    - Unconnected trees

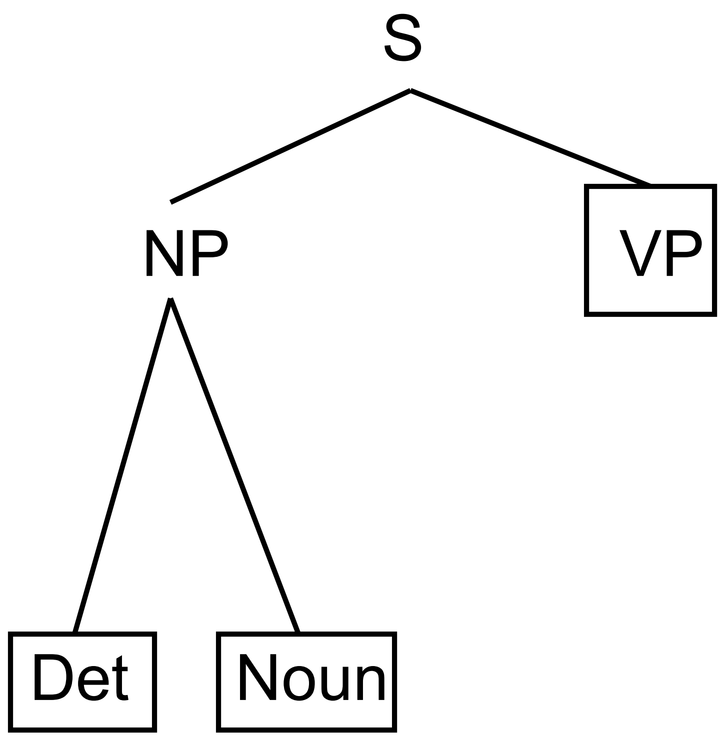
# Top-down - example

S

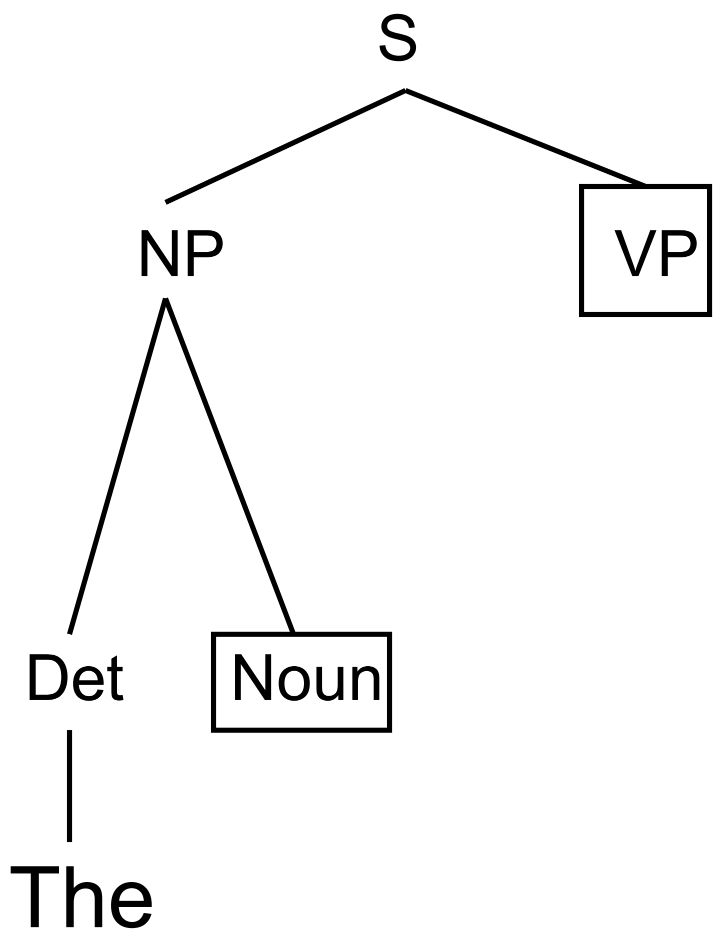
# Top-down - example



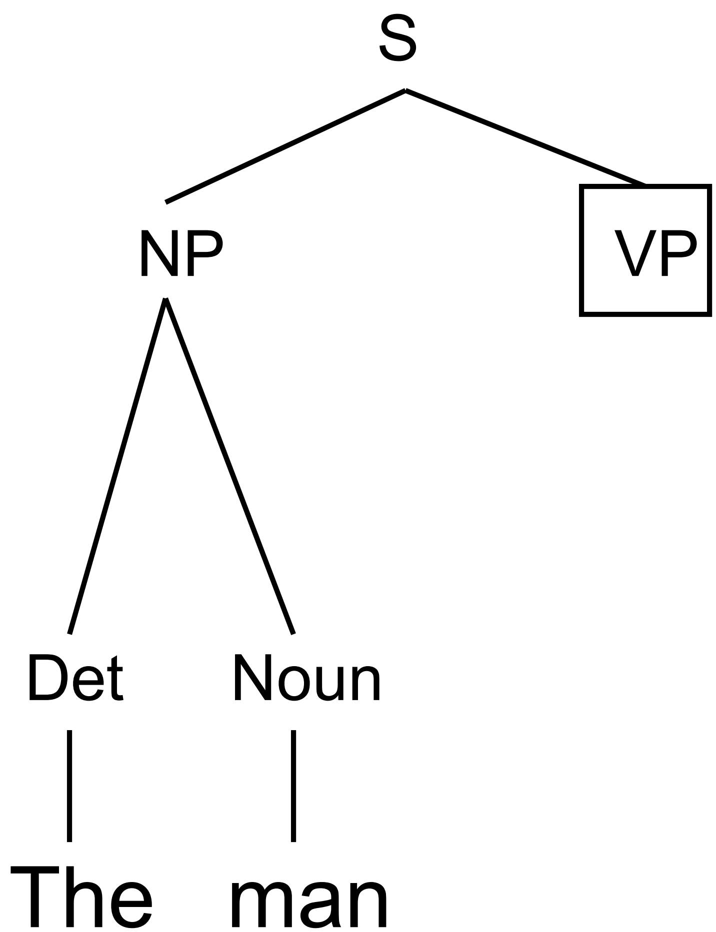
# Top-down - example



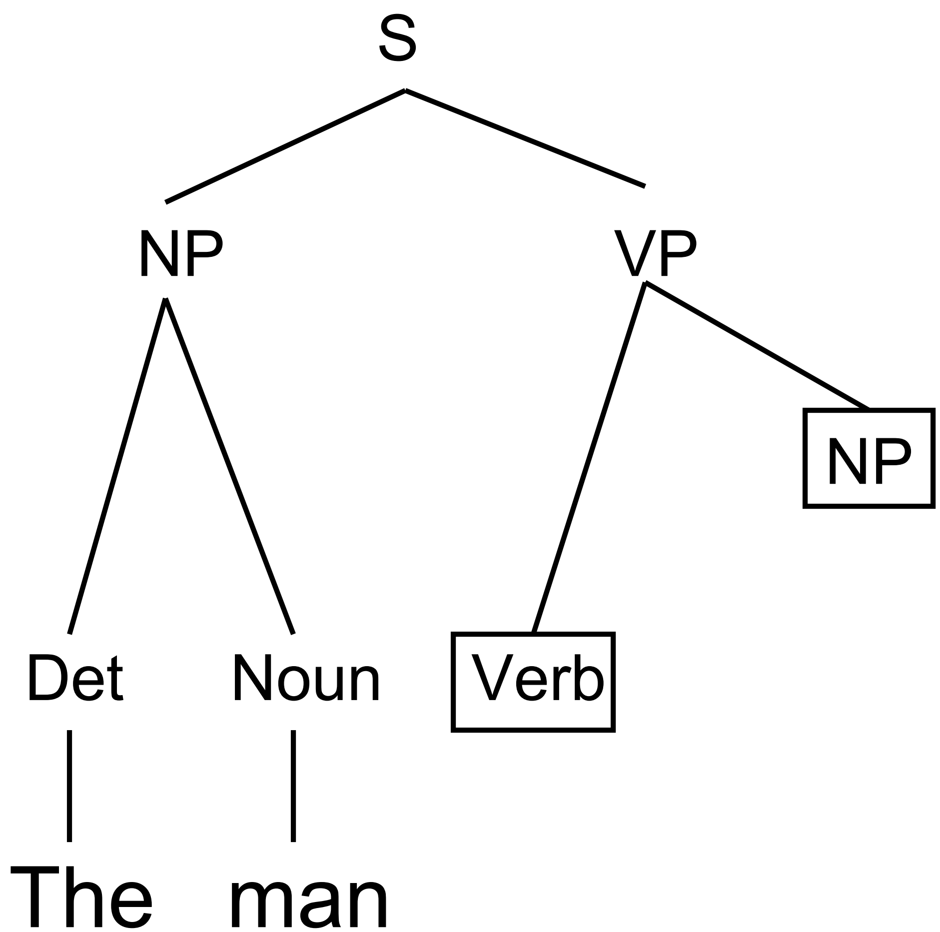
# Top-down - example



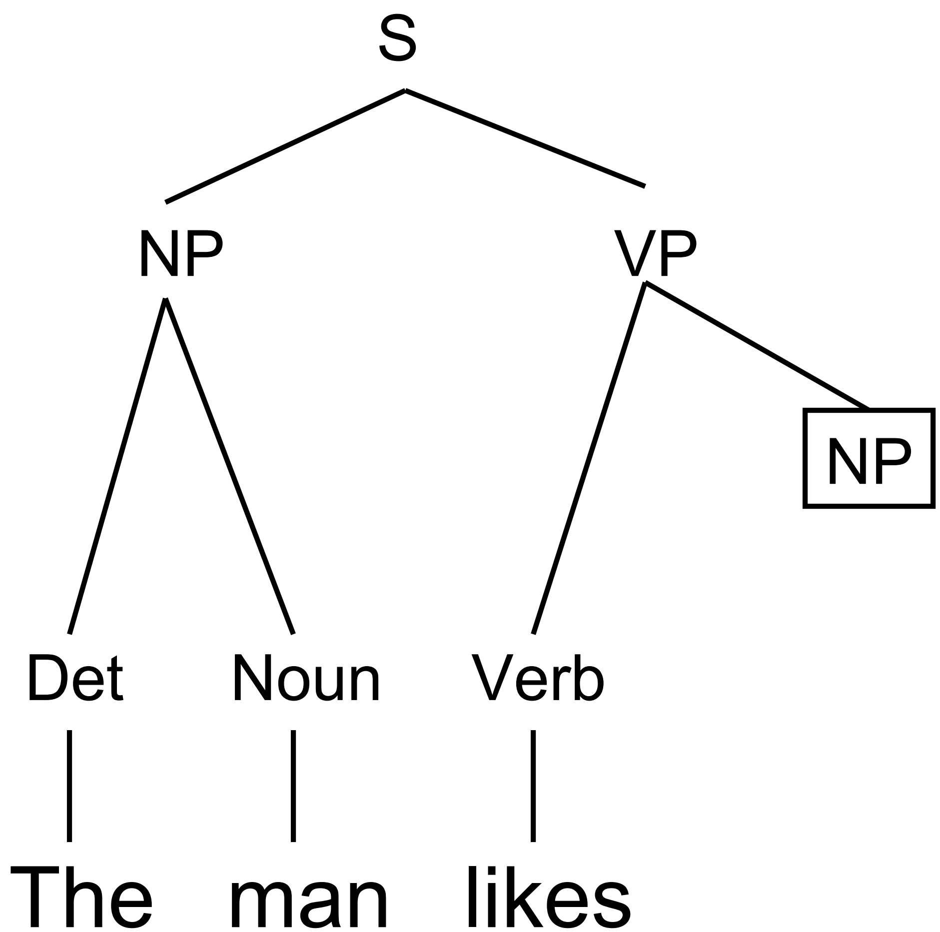
# Top-down - example



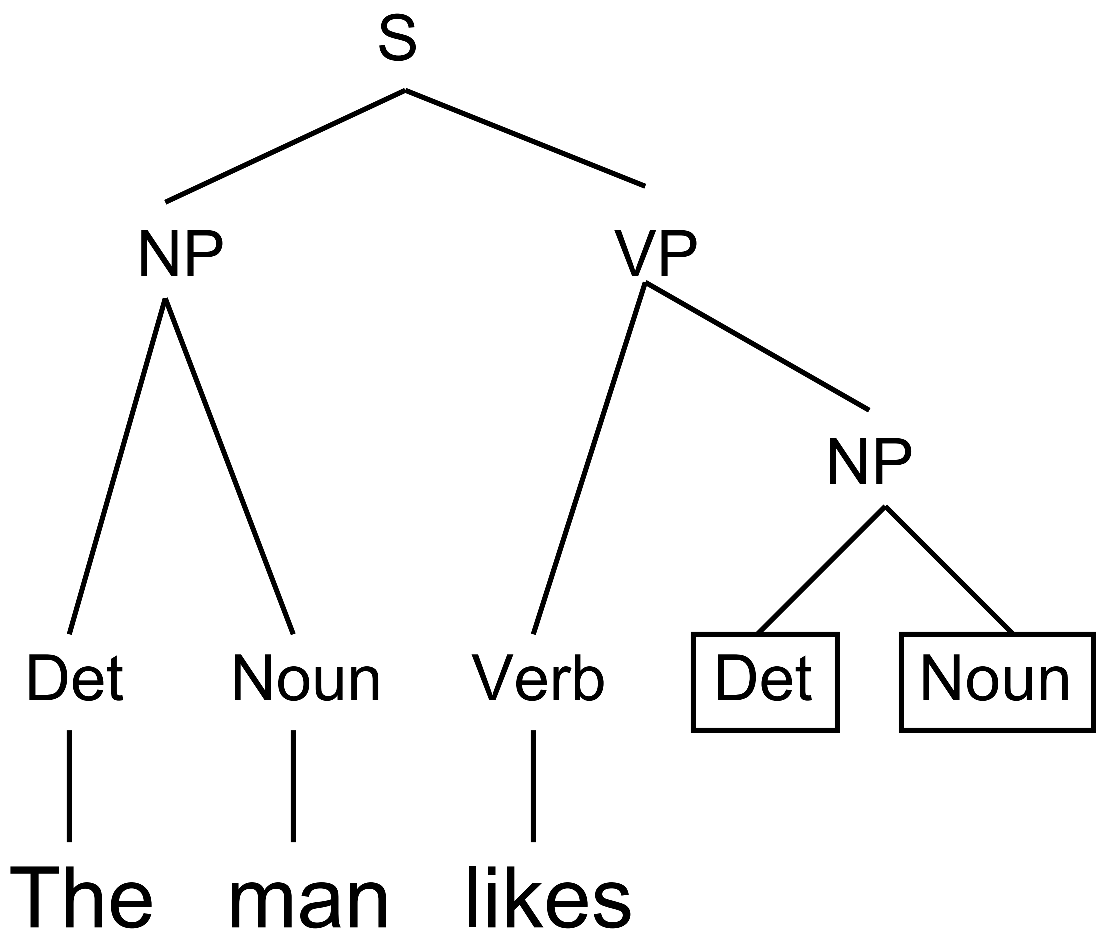
# Top-down - example



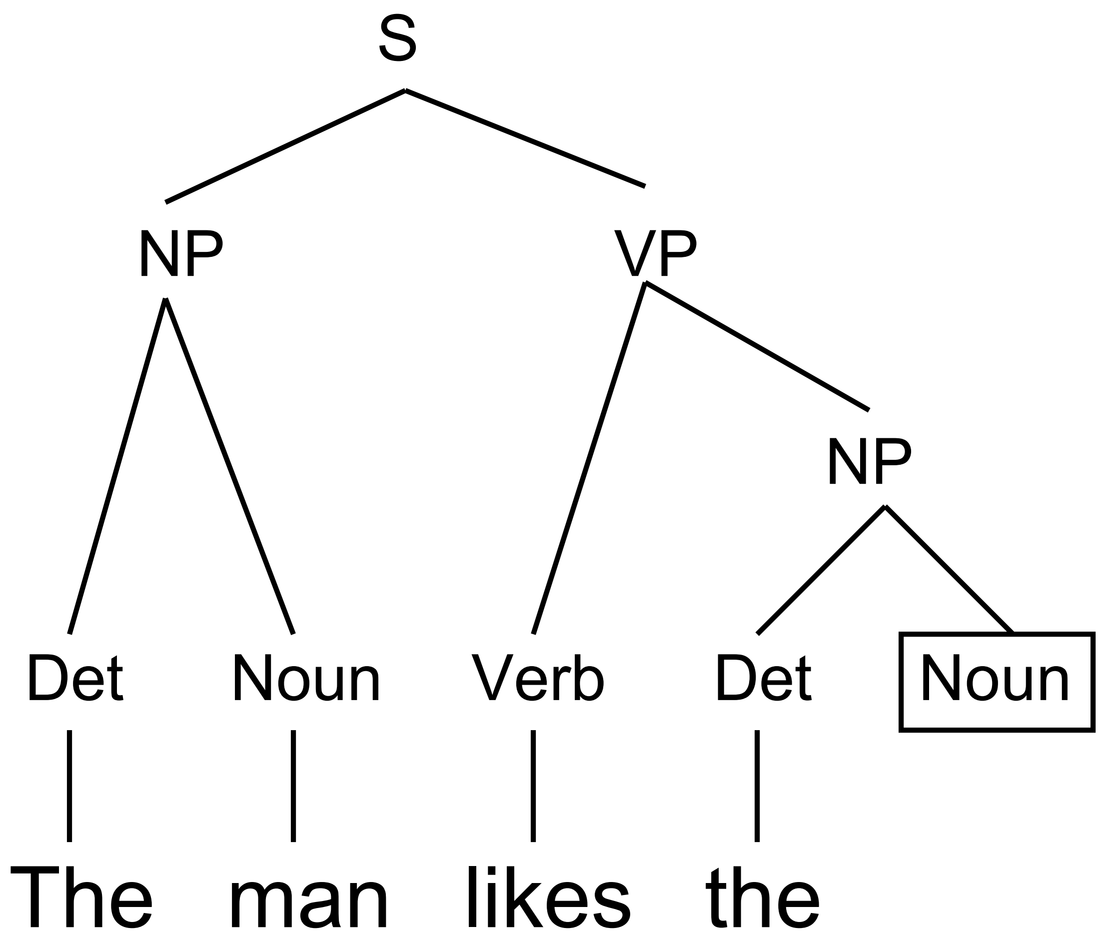
# Top-down - example



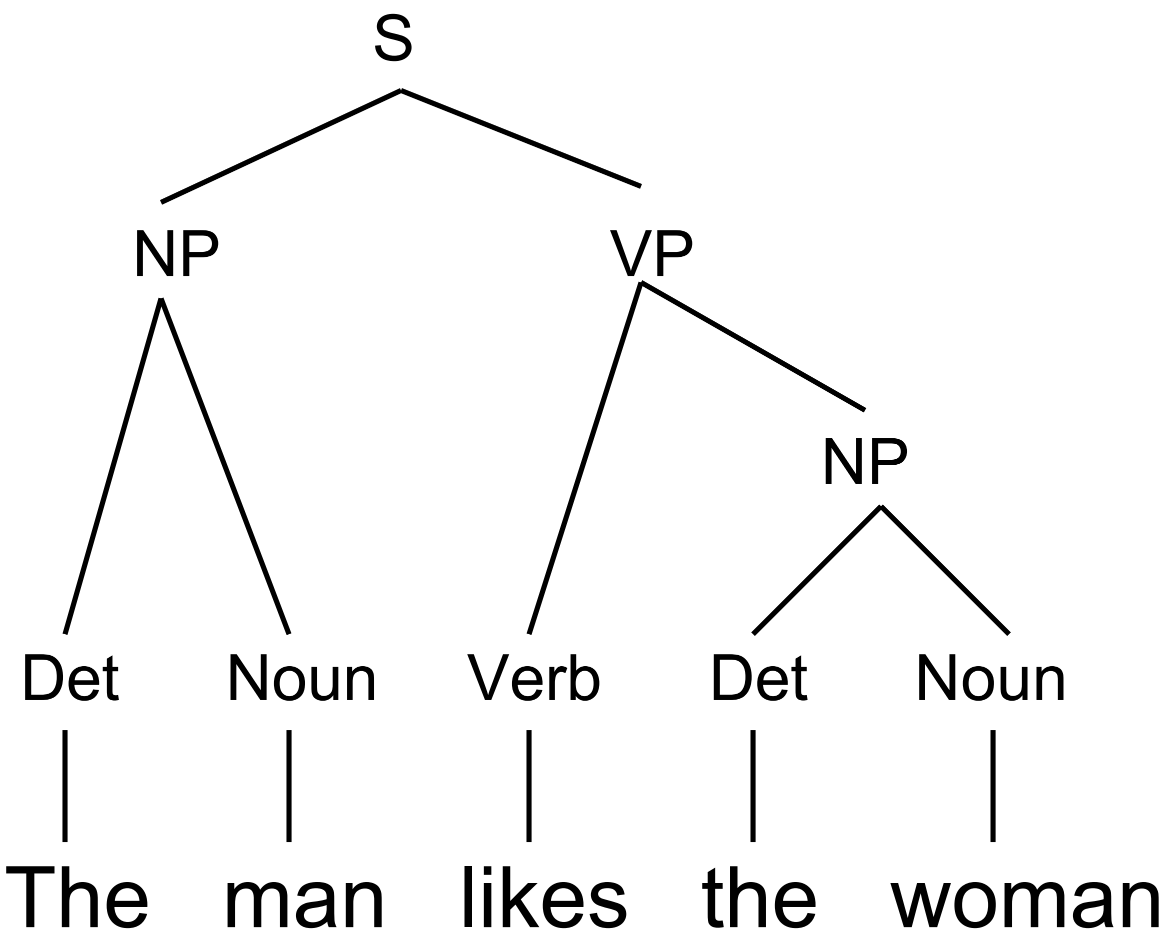
# Top-down - example



# Top-down - example



# Top-down - example



# Top-down - evaluation

- Advantage:
  - Starts with S-node, therefore never tries to form a structure that will never form an S
- Disadvantages:
  - Can try to build trees inconsistent with input (if we had rule VP $\rightarrow$ V PP; or VP $\rightarrow$ V)
  - Left-recursion

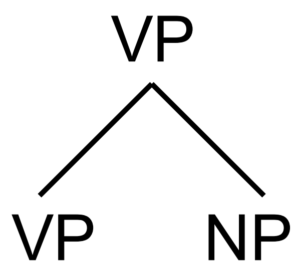
# Top-down - evaluation

- Left-recursion, cont'd:
  - Left-recursion = some LHS can be expanded through series of rules such that left corner of one of these expansions is in same LHS category
  - VP → VP NP
- Parser gets caught in endless loop

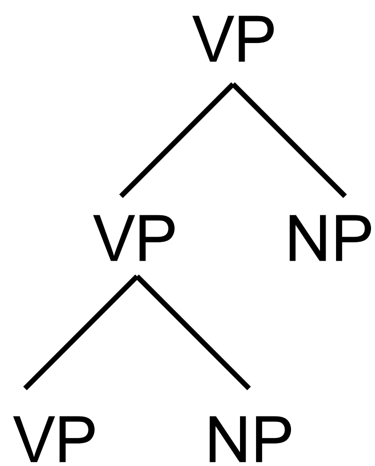
# Top-down - left-recursion

VP

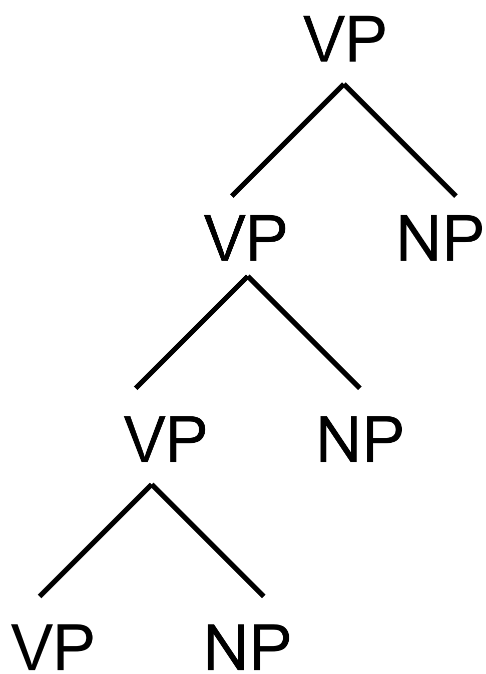
# Top-down - left-recursion



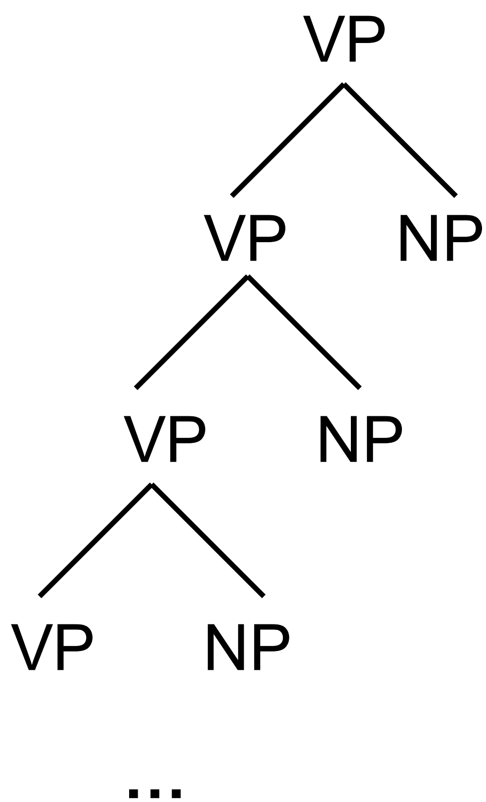
# Top-down - left-recursion



# Top-down - left-recursion



# Top-down - left-recursion



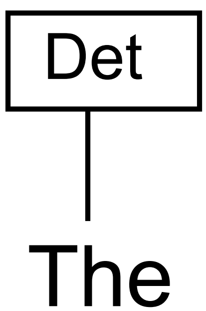
# Bottom-up - intro

- Look at input, then try to find rules in grammar that apply to input
- Stack keeps track of what has been found so far and still needs to be integrated in parse tree

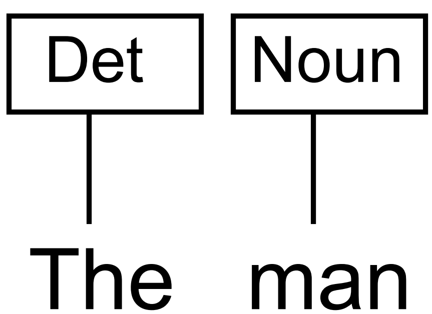
# Bottom-up - intro

- **shift** = push categories on stack that still need to be integrated
- **reduce** = apply grammar rules to categories in stack
- Shift-reduce parser

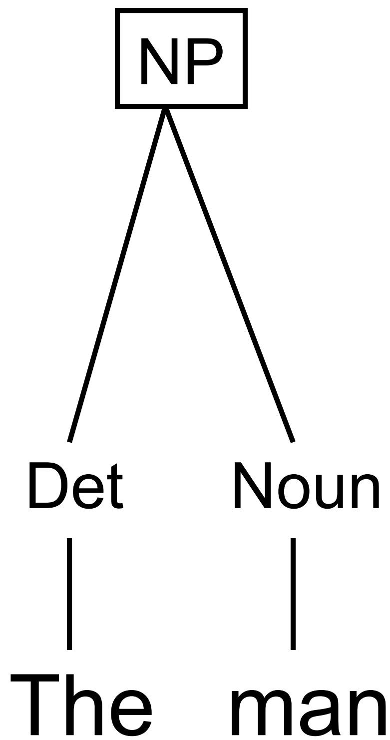
# Bottom-up - example



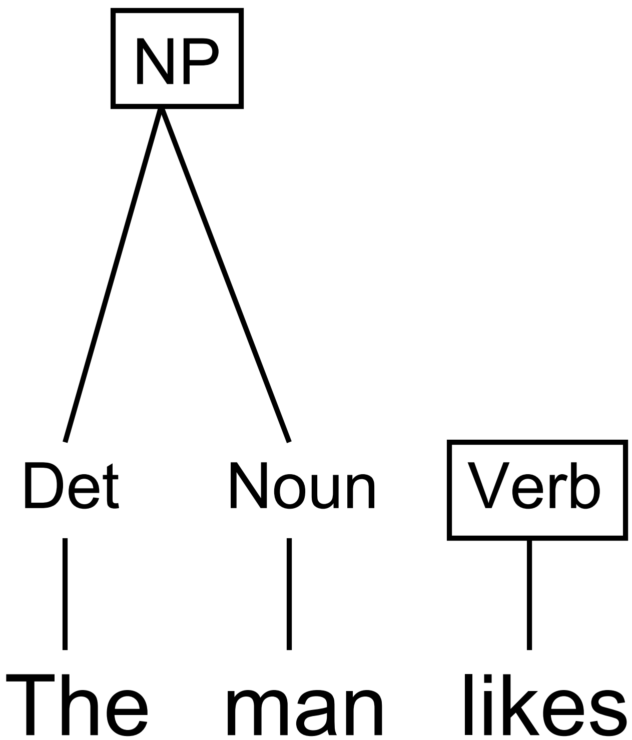
# Bottom-up - example



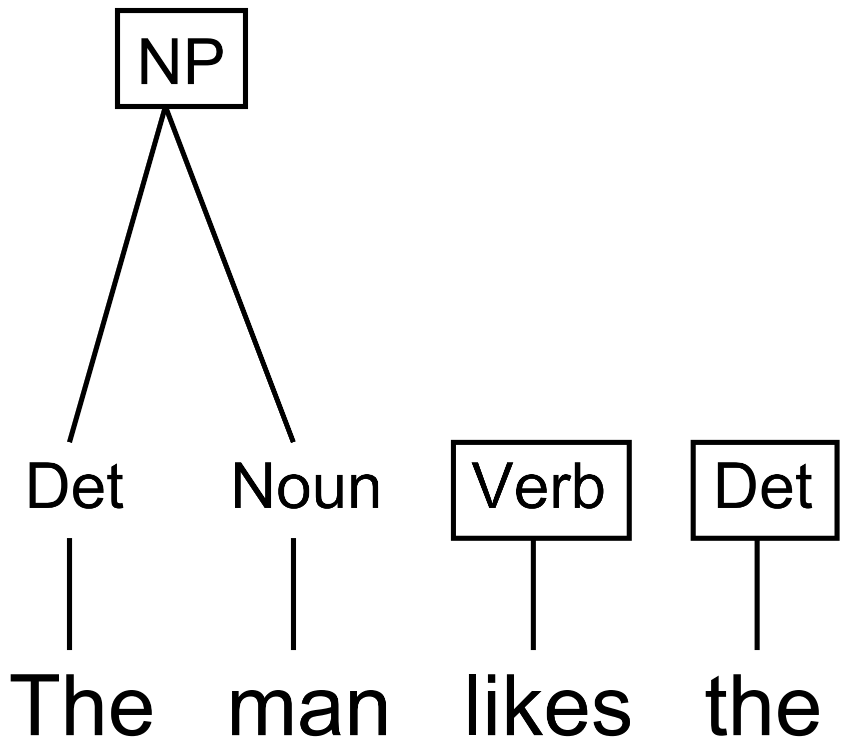
# Bottom-up - example



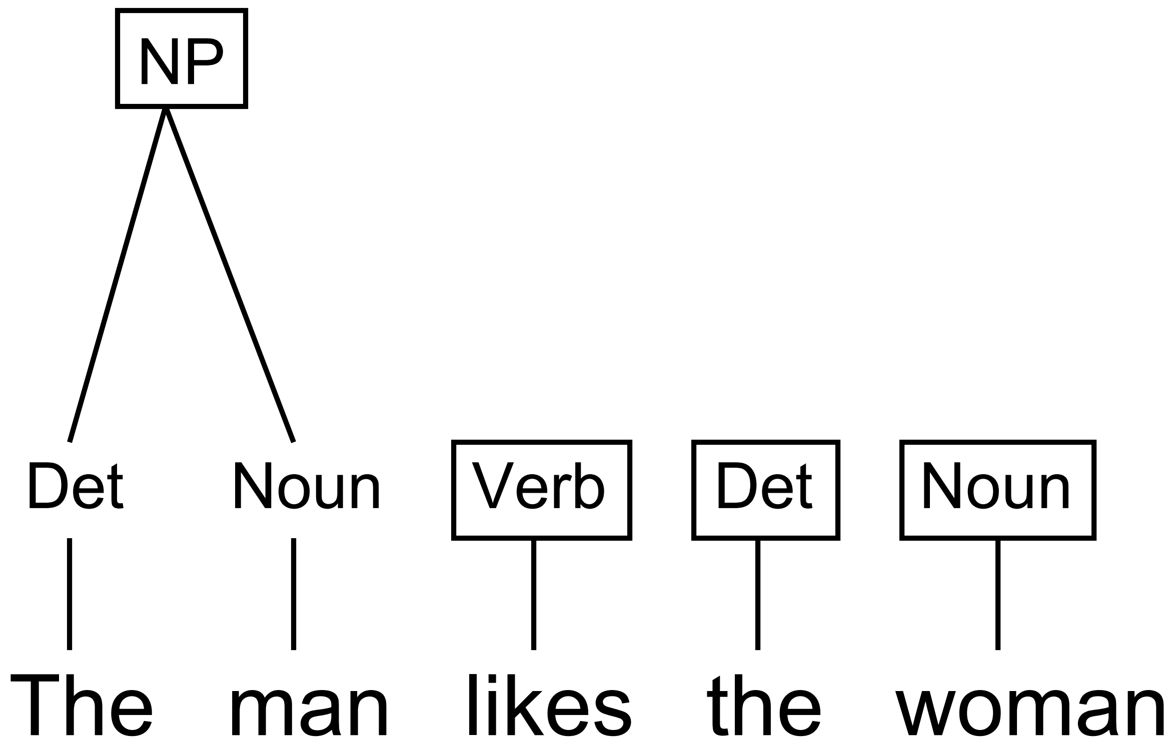
# Bottom-up - example



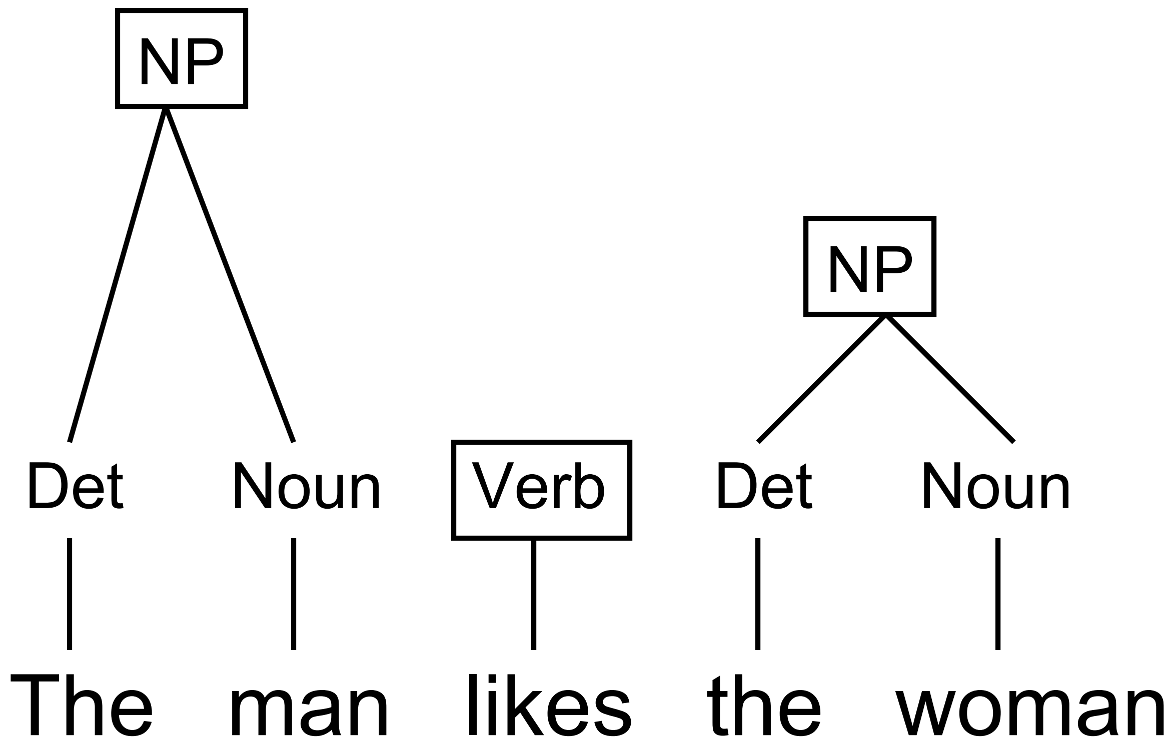
# Bottom-up - example



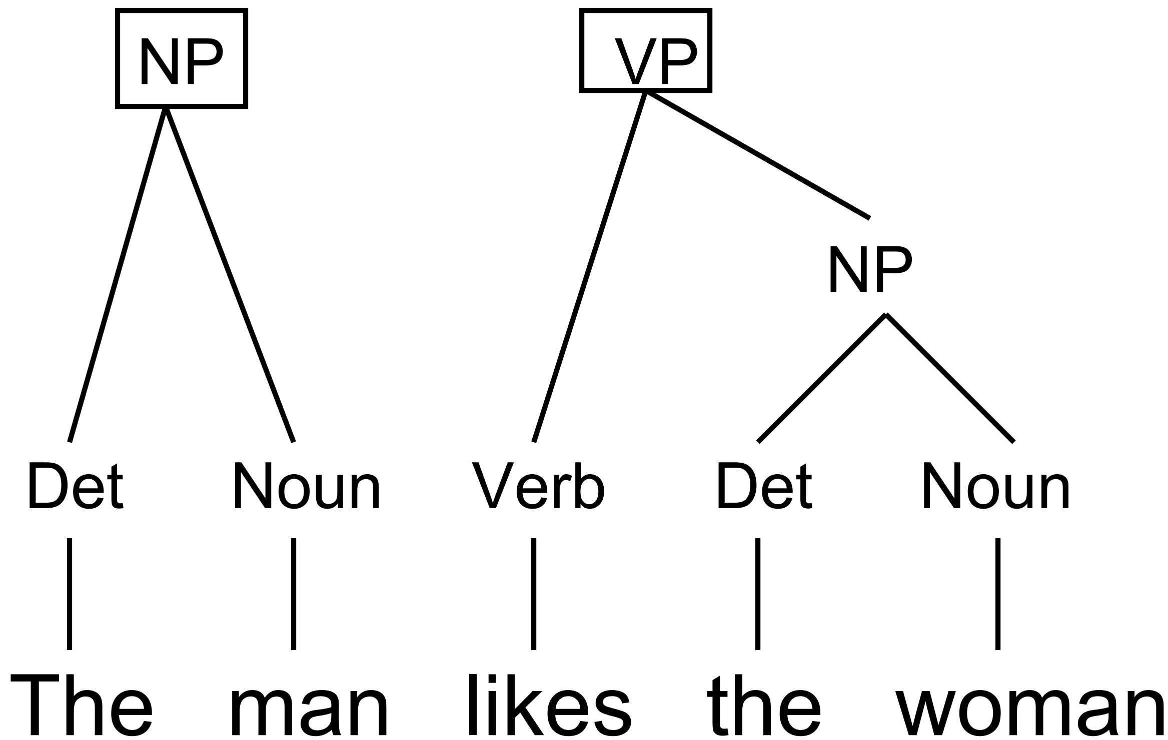
# Bottom-up - example



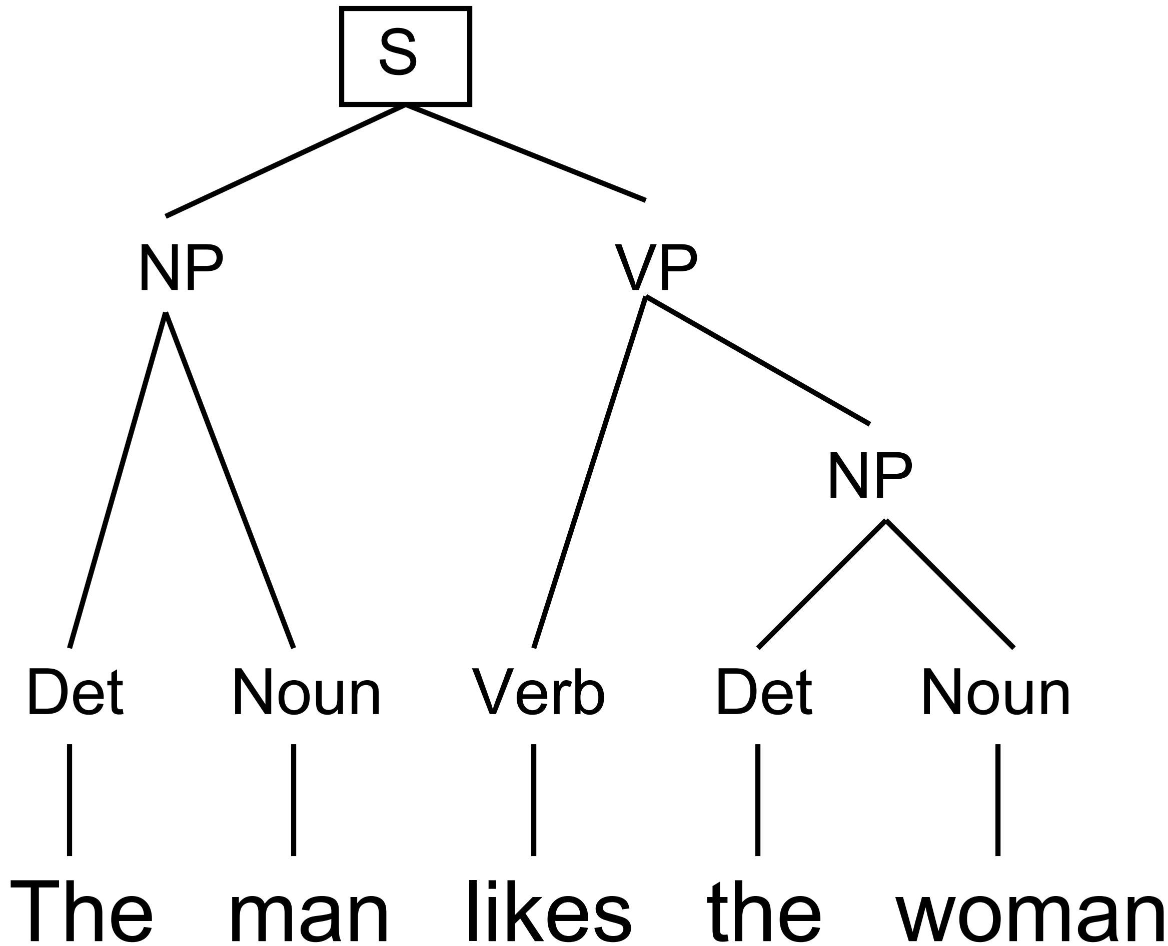
# Bottom-up - example



# Bottom-up - example



# Bottom-up - example



# Bottom-up - evaluation

- Advantages:
  - Does not predict trees that are inconsistent with input
  - Can handle left-recursion
- Disadvantage:
  - Can generate structures that never result in an S

# Top-down vs. bottom-up

- Top-down
  - Good
    - Always produces S
  - Bad
    - Can generate structures inconsistent w/ input
    - No left-recursion
- Bottom-up
  - Good
    - Always generates structures consistent w/ input
    - Handles left-recursion
  - Bad
    - Can generate non-S structures

# Memory complexity: Tree geometries

- Three tree-geometric kinds of sentence structures:

  - **Left-branching**: easy to process, and common in a head-final language (but somewhat rare in a head-initial language like English, although still easy to process):

    *John's brother's neighbor's friend's dog's tail fell off.*

    Rules: NP → NP-Gen N ;  NP-Gen → NP 's

  - **Right-branching**: easy to process, and common in a head-initial like English:

    *The dog chased the cat that caught the mouse that ate the cheese that was on the counter.*

  - **Center-embedded**: hard to process.

    *# The mouse that the cat that the dog chased caught squeaked.*

# Processing complexity

- One component of processing complexity: the storage space that is needed to parse a sentence

- The more unfinished stuff you have to store, the harder parsing becomes

# Memory complexity: Stack depth

- One component of the memory complexity of a parsing algorithm: Its maximal stack-depth when parsing a sentence

- Top-down
  - Bounded stack depth on left-branching structures
  - Unbounded stack depth on right-branching structures and center-embedded structures

- Bottom-up
  - Bounded stack depth on right-branching structures
  - Unbounded stack depth on left-branching structures and center-embedded structures

- Neither is a good model of human processing complexity

# Left-corner - intro

- Rule only predicted (top-down) if current input (bottom-up) matches leftmost corner (left-corner) of the RHS of a rule
  - VP → Verb NP
- Stack keeps track of what input is still needed to complete a predicted rule

# Left-corner - example

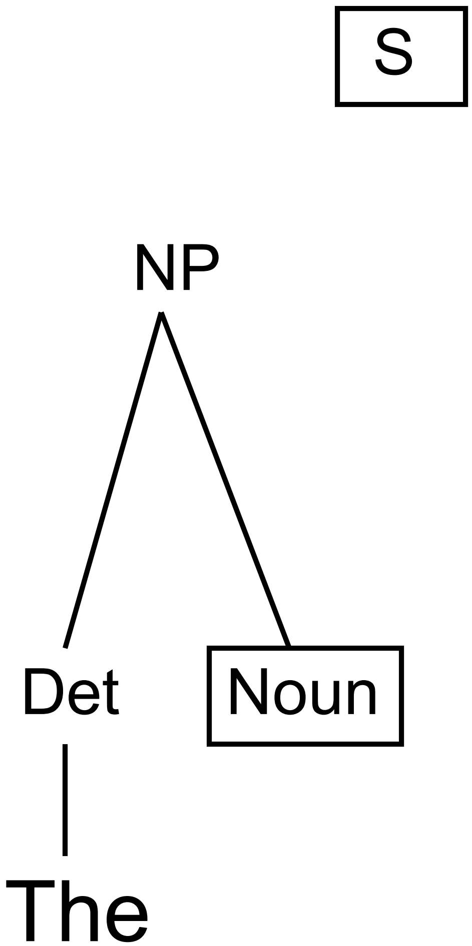
$\boxed{S}$

Det

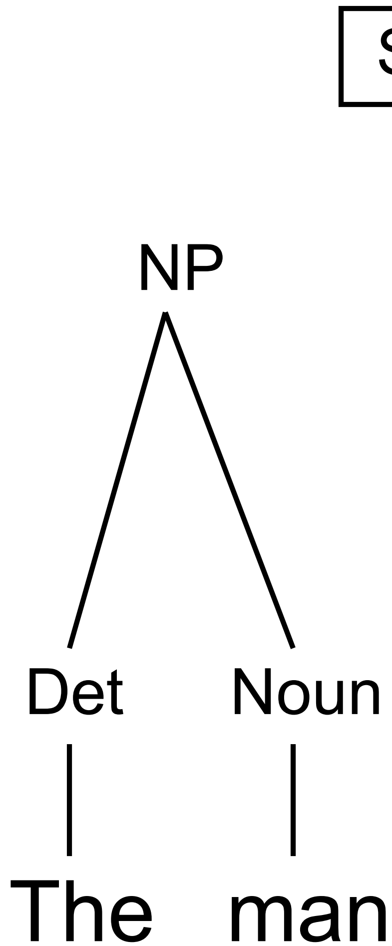
|

The

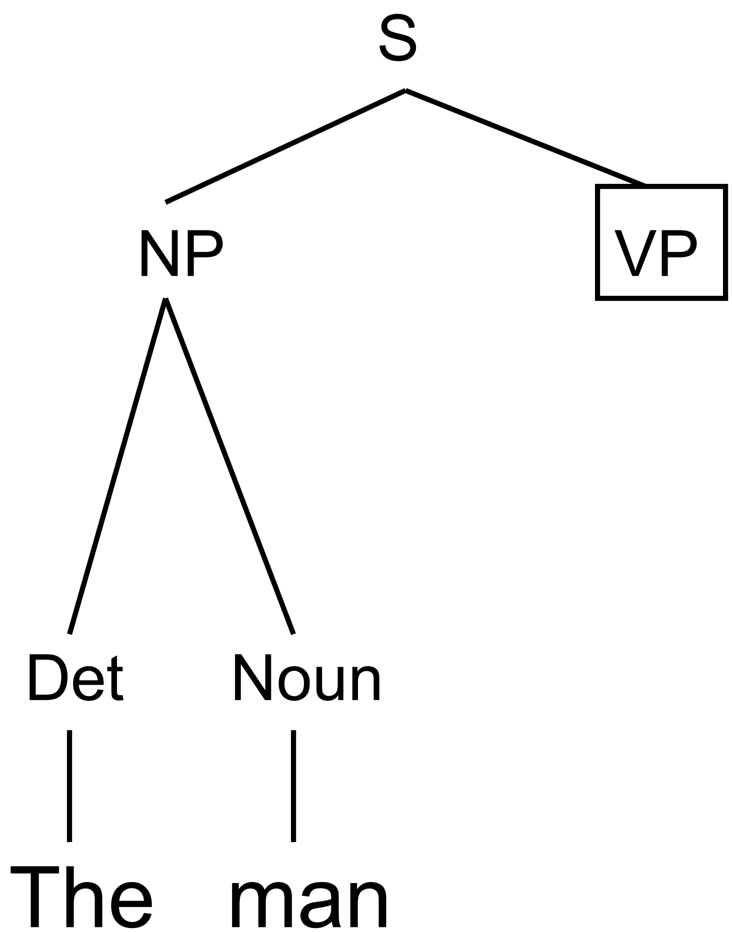
# Left-corner - example



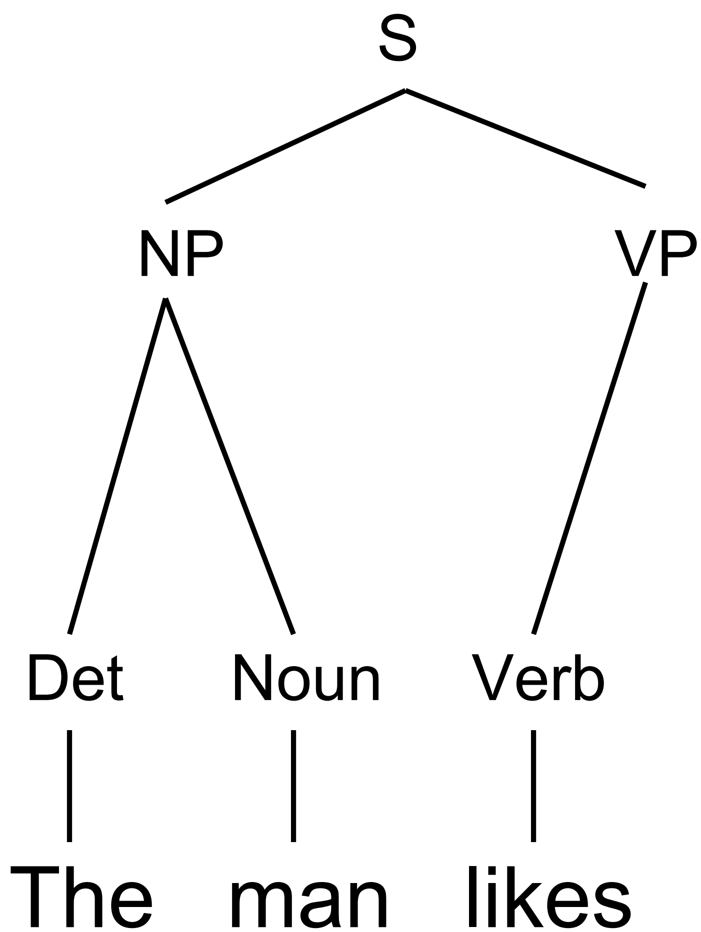
# Left-corner - example



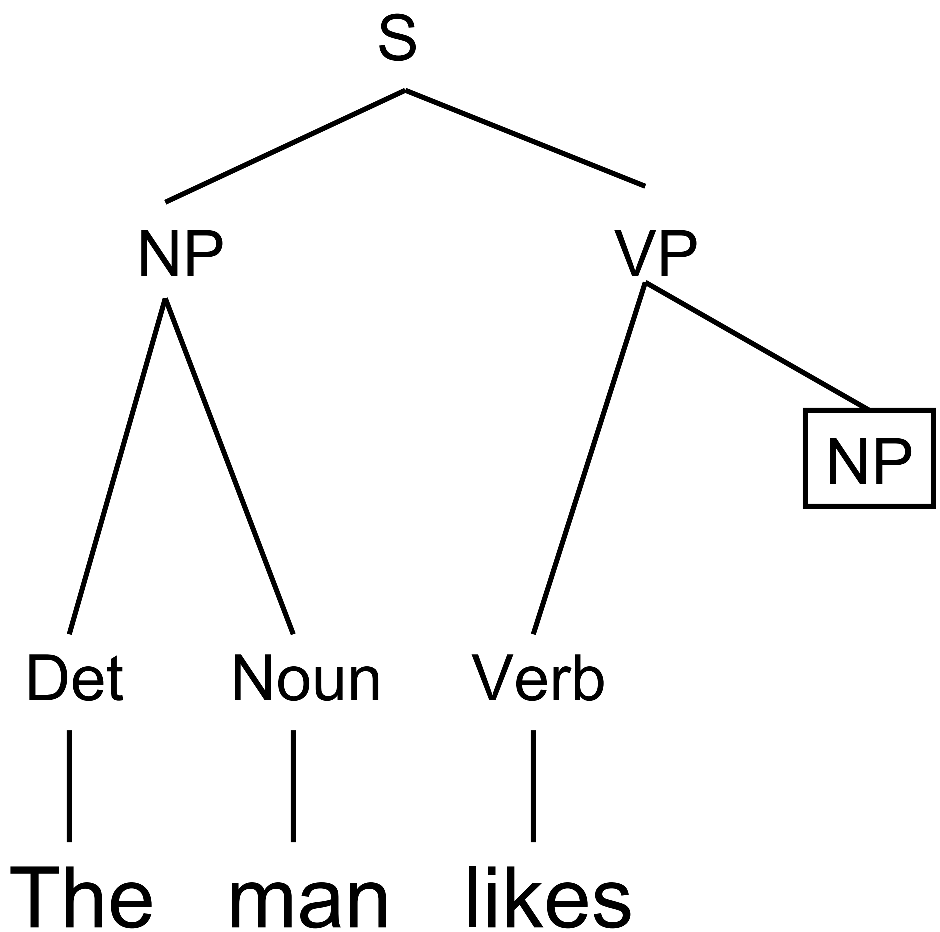
# Left-corner - example



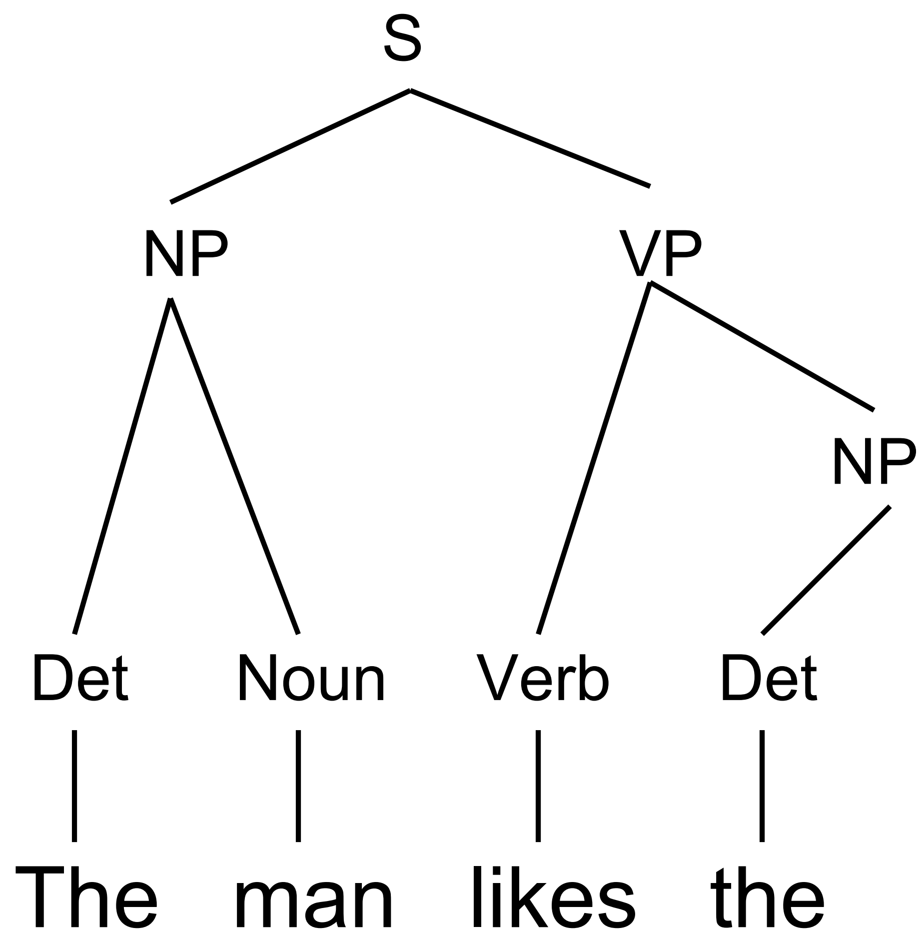
# Left-corner - example



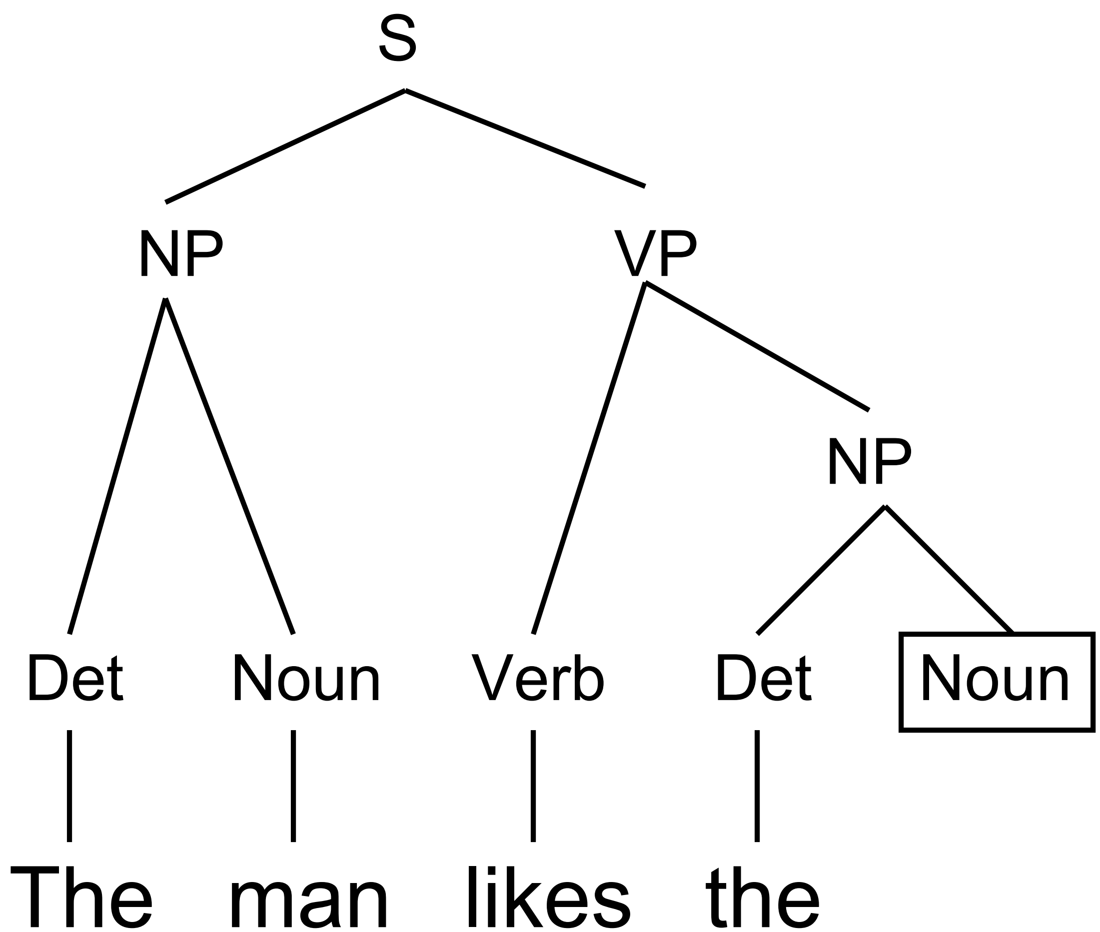
# Left-corner - example



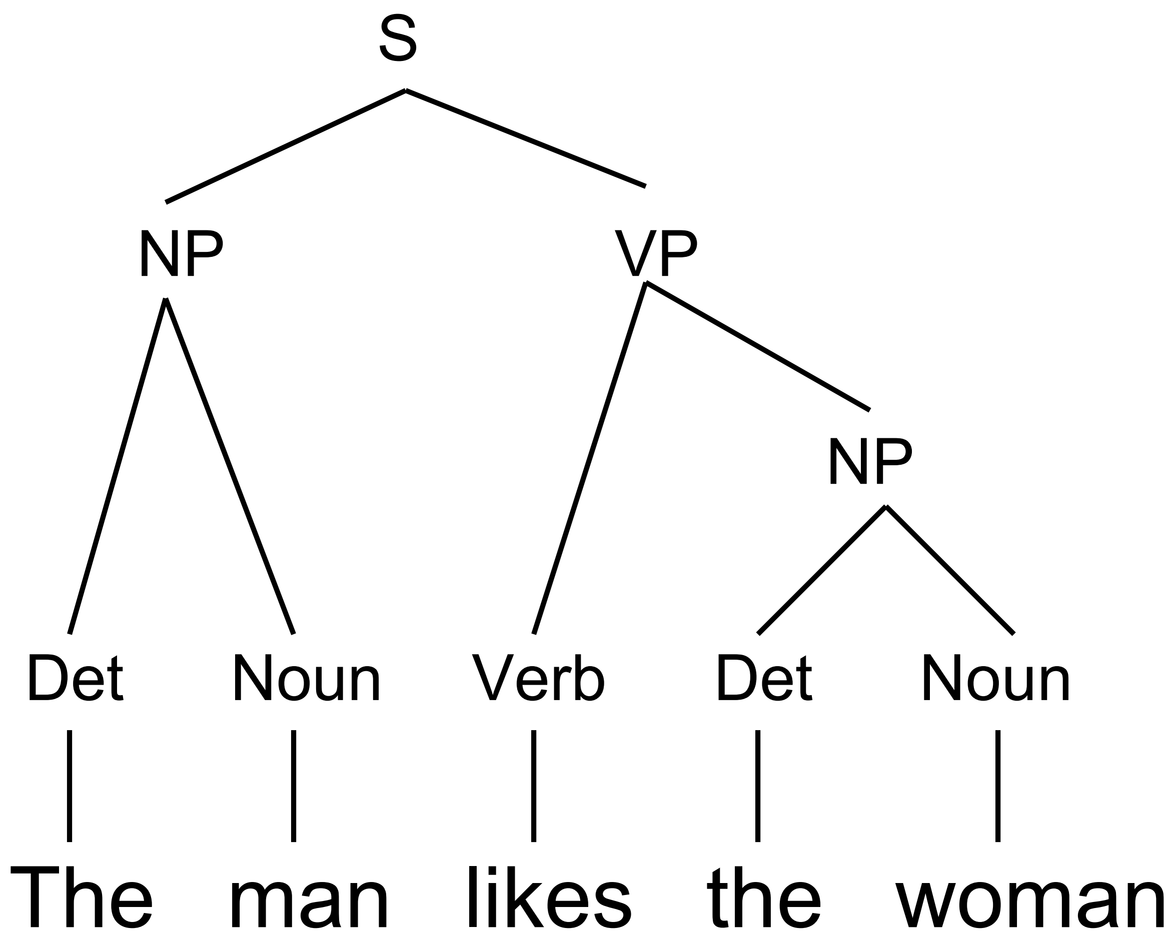
# Left-corner - example



# Left-corner - example



# Left-corner - example



# Memory complexity: Stack depth

- One component of the memory complexity of a parsing algorithm: Its maximal stack-depth when parsing a sentence

- Left-corner
  - Bounded stack depth on left- and right-branching structures
  - Unbounded stack depth on center-embedded structures

- A better basis for a good model of human processing complexity (not sufficient, but better as a basis)

# Processing complexity

| | sentence structure | | |
|---|---|---|---|
| | **LB** | **CE** | **RB** |
| **TD stack size** | Unbounded | Unbounded | Bounded |
| **BU stack size** | Bounded | Unbounded | Unbounded |
| **LC stack size** | Bounded | Unbounded | Bounded |
| **humans** | Easy | Hard | Easy |

# Chart parsing - intro

- Still a problem: ambiguity

  *The tall man with brown hair saw the short woman with blond hair*.
- *"saw"* is ambiguous
- *The tall man with brown hair* has to be parsed twice, for Noun and Verb meaning
- In realistic applications: enormous efficiency problem

# Chart parsing - intro

- Solution: give the parser a memory!
- Keep track of partially and completely parsed rules from grammar
- Look up parsed rules instead of reparsing them
- **chart** = memory for parser

# Chart parsing - definitions

- Chart:
  - **edges** = rules from the grammar
  - **incomplete / active edges** = partially parsed edges
  - **complete / inactive edges** = completely parsed edges
  - Once an edge is entered into chart, it stays there

# Chart parsing - edges

- Information about edges in chart:
  - Syntactic category of the edge (e.g. *NP*)
  - Where in the sentence the edge begins (left end)
  - Where in the sentence the edge ends (right end)
  - Pointers to further inactive edges (e.g. to *Det Noun* for *NP*)
  - For active edges: list of what categories are still needed to complete the edge (make it inactive)

# Chart parsing - edges

- Edge notation:
  - [category] / [what's there] **.** [what's needed]
- Examples of edges:
  - S / NP **.** VP
  - NP / Det Noun **.**

# Chart parsing - edges

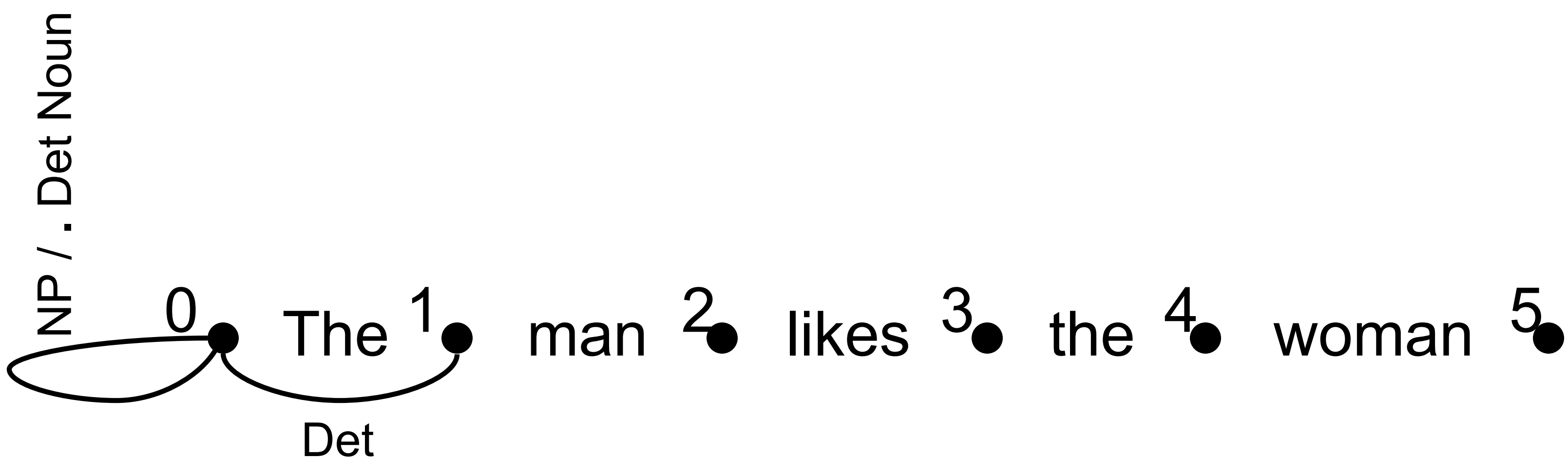
- Edge notation:
  - [category] / [what's there] **.** [what's needed]
- Examples of edges:
  - S / NP **.** VP
  - NP / Det Noun **.**

# Chart parsing - cont'd

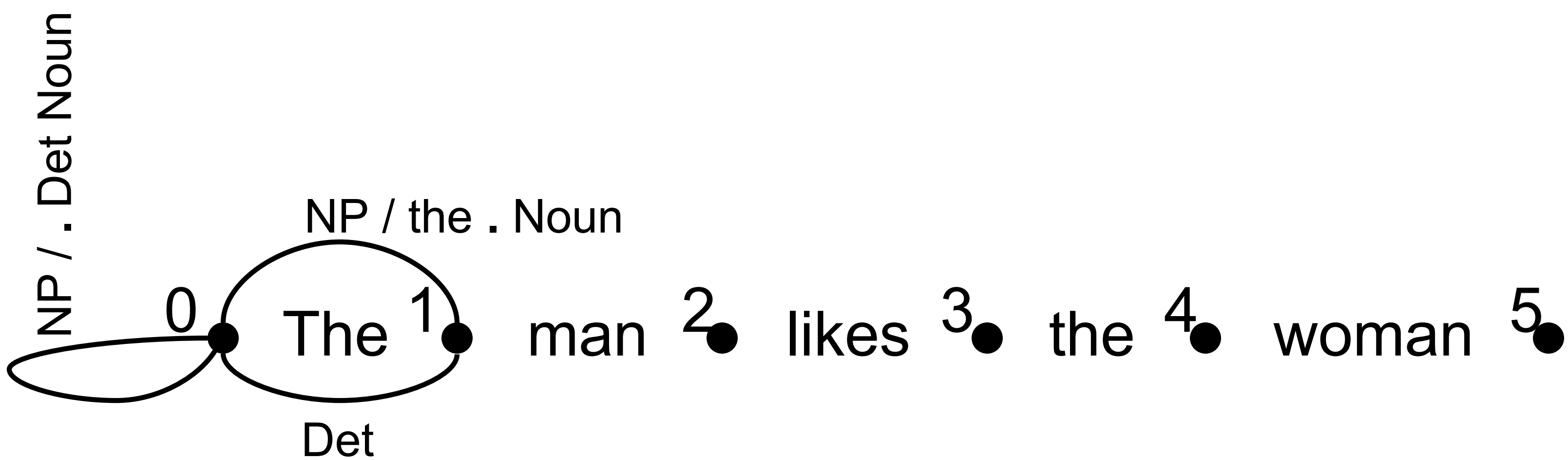
- Fundamental Rule
  - Look for matches between categories needed in an active edge and set of inactive edges
- Top-down
  - Initialize chart w/ empty active *S* edge
- Bottom-up
  - Initialize chart w/ inactive edges for words

# Chart parsing - example

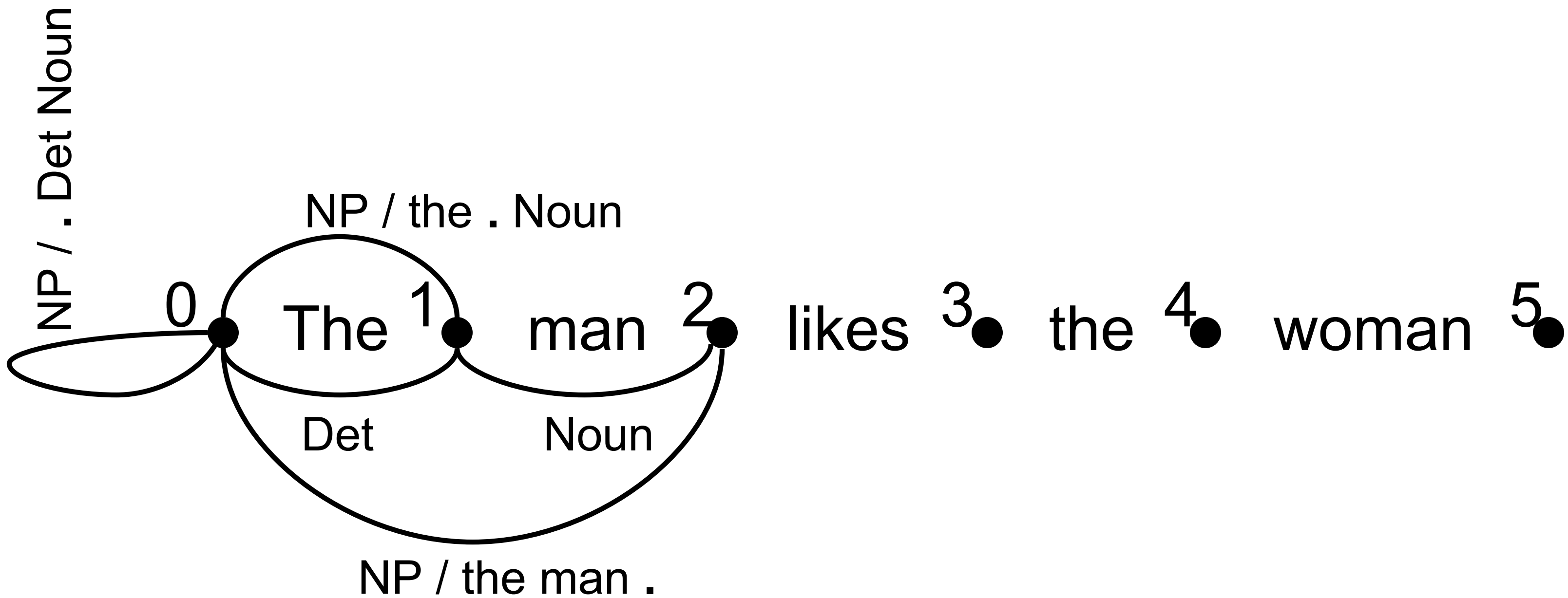
# Chart parsing - example



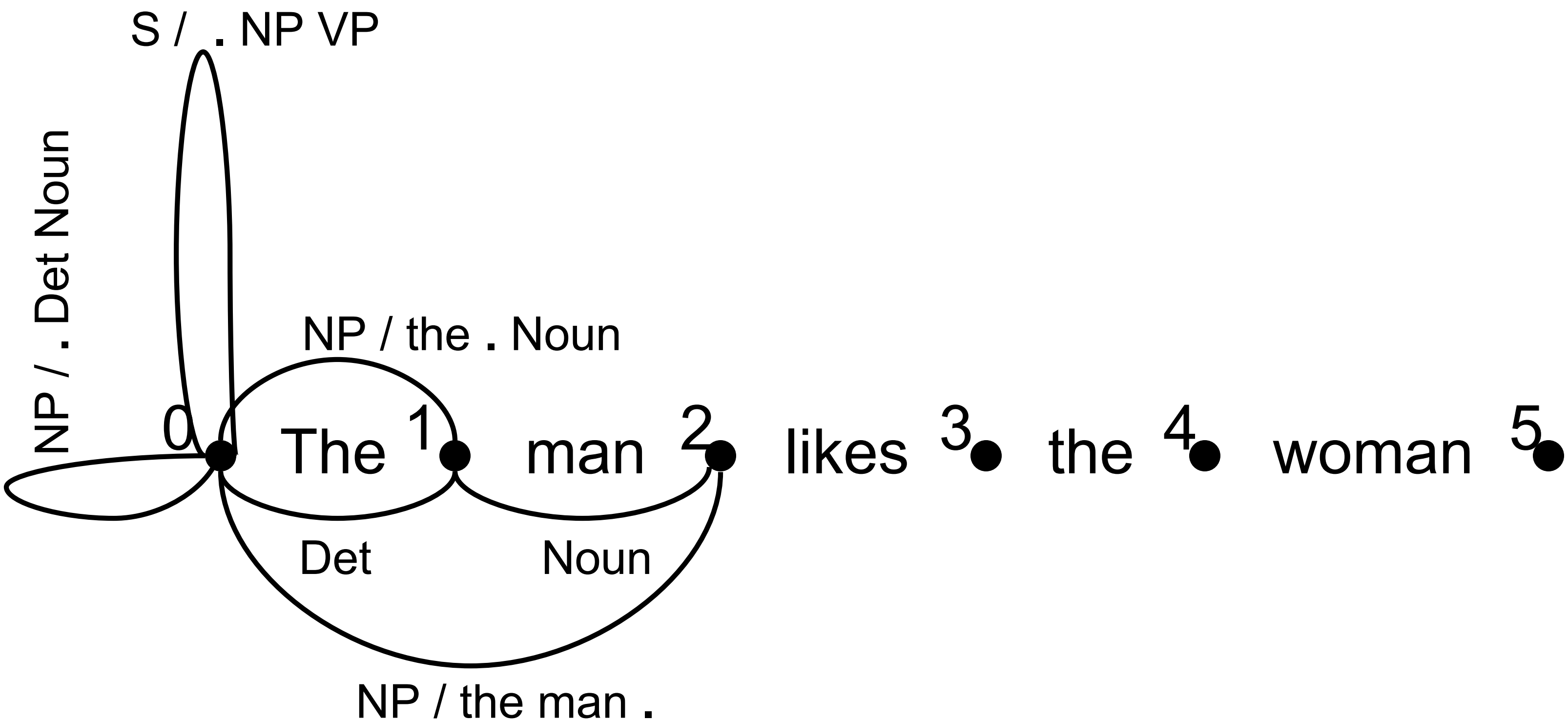
# Chart parsing - example



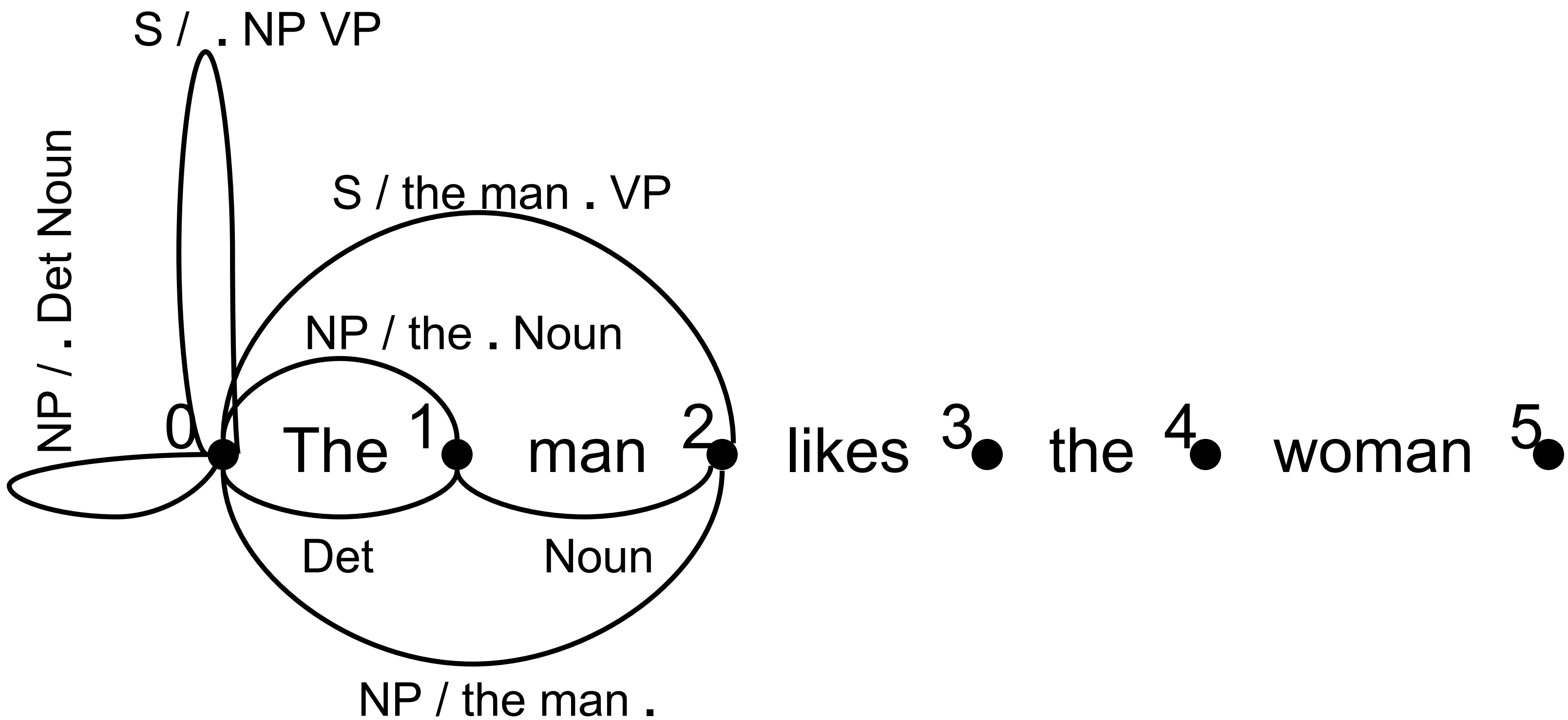
# Chart parsing - example



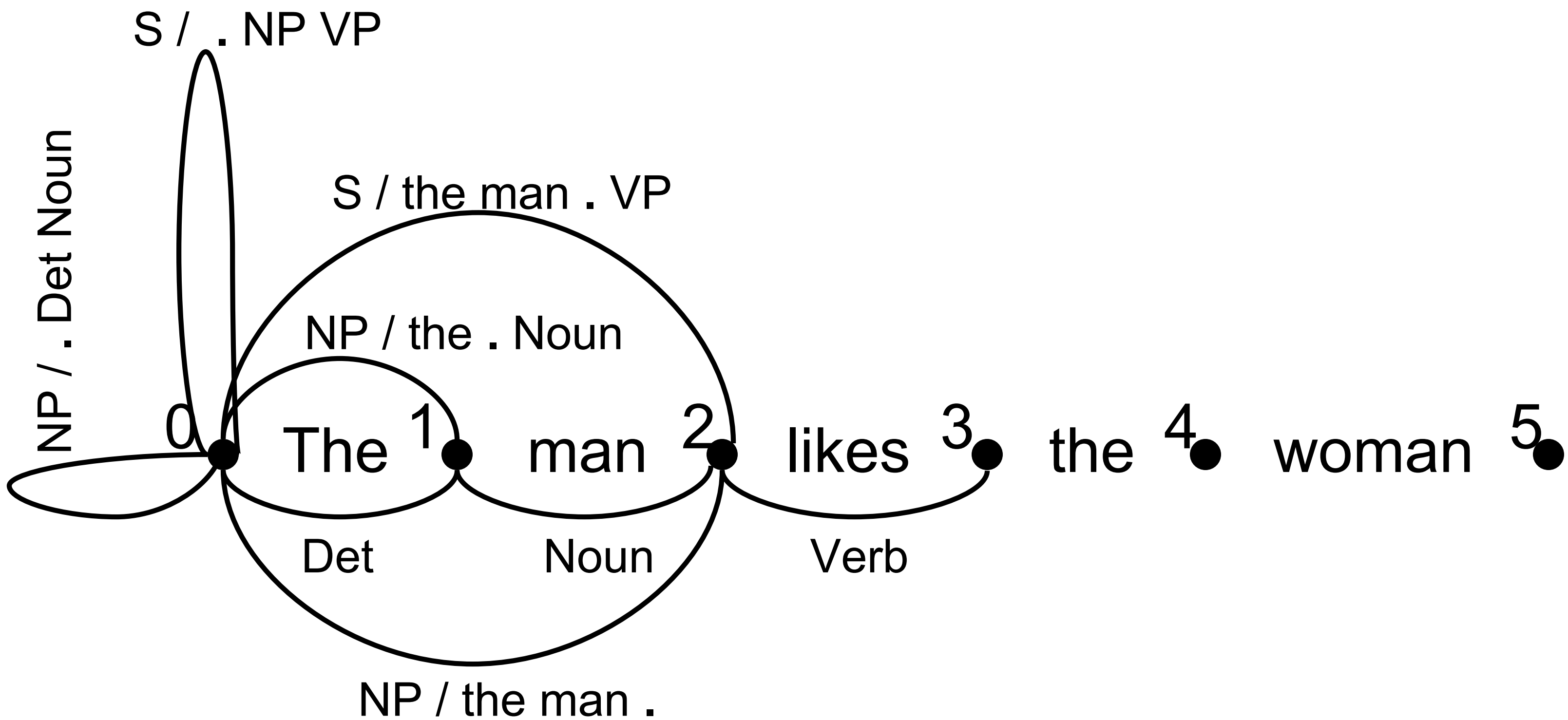
# Chart parsing - example



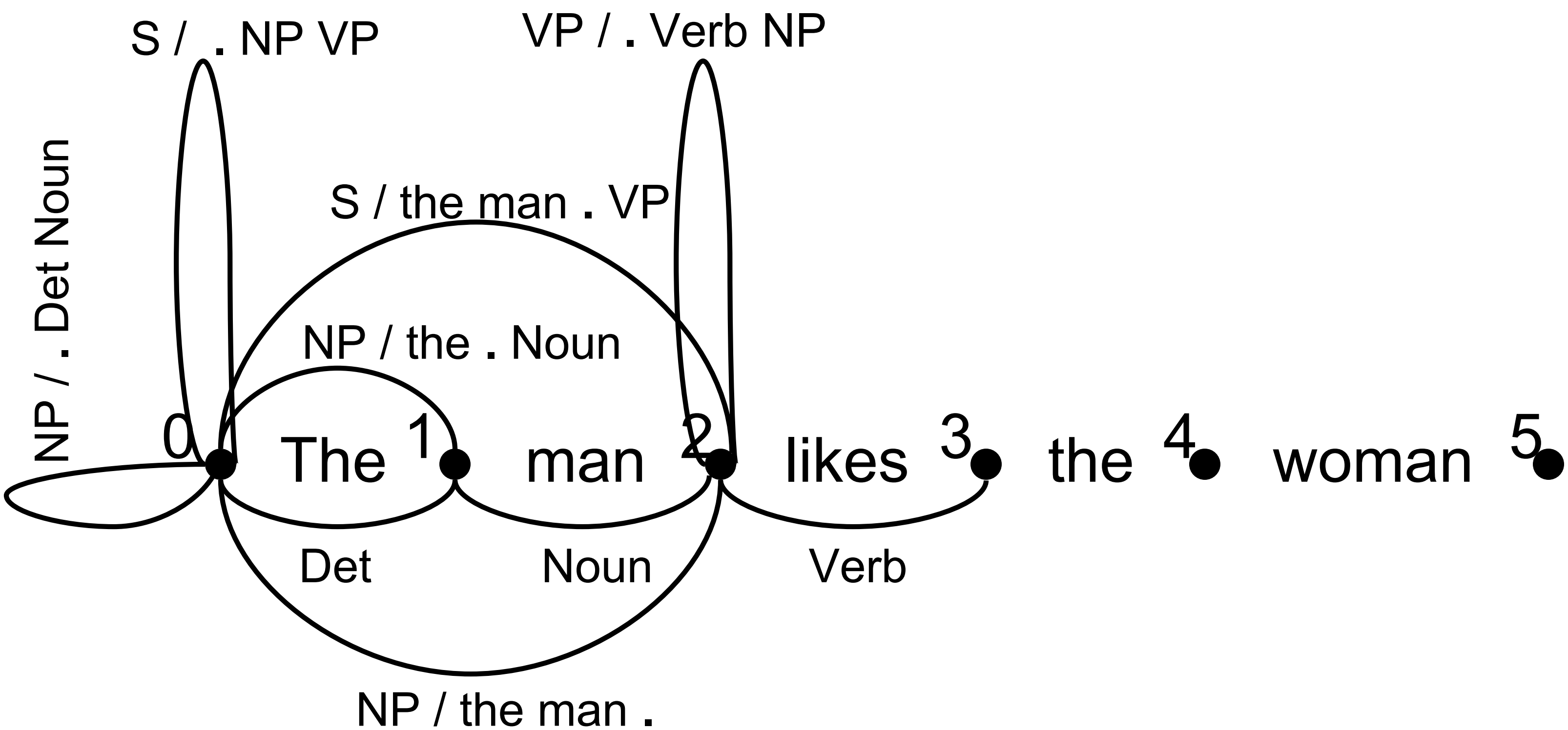
# Chart parsing - example



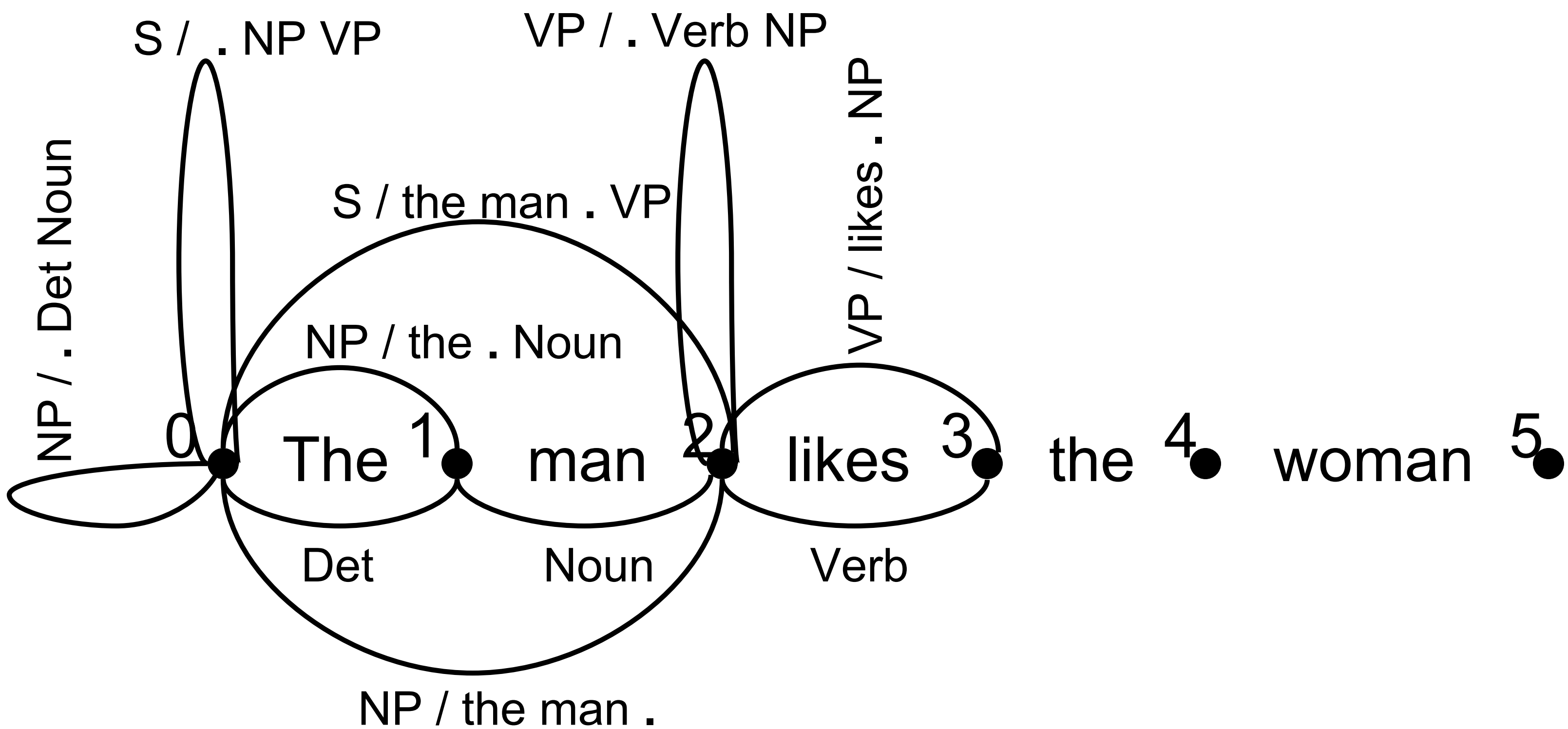
# Chart parsing - example



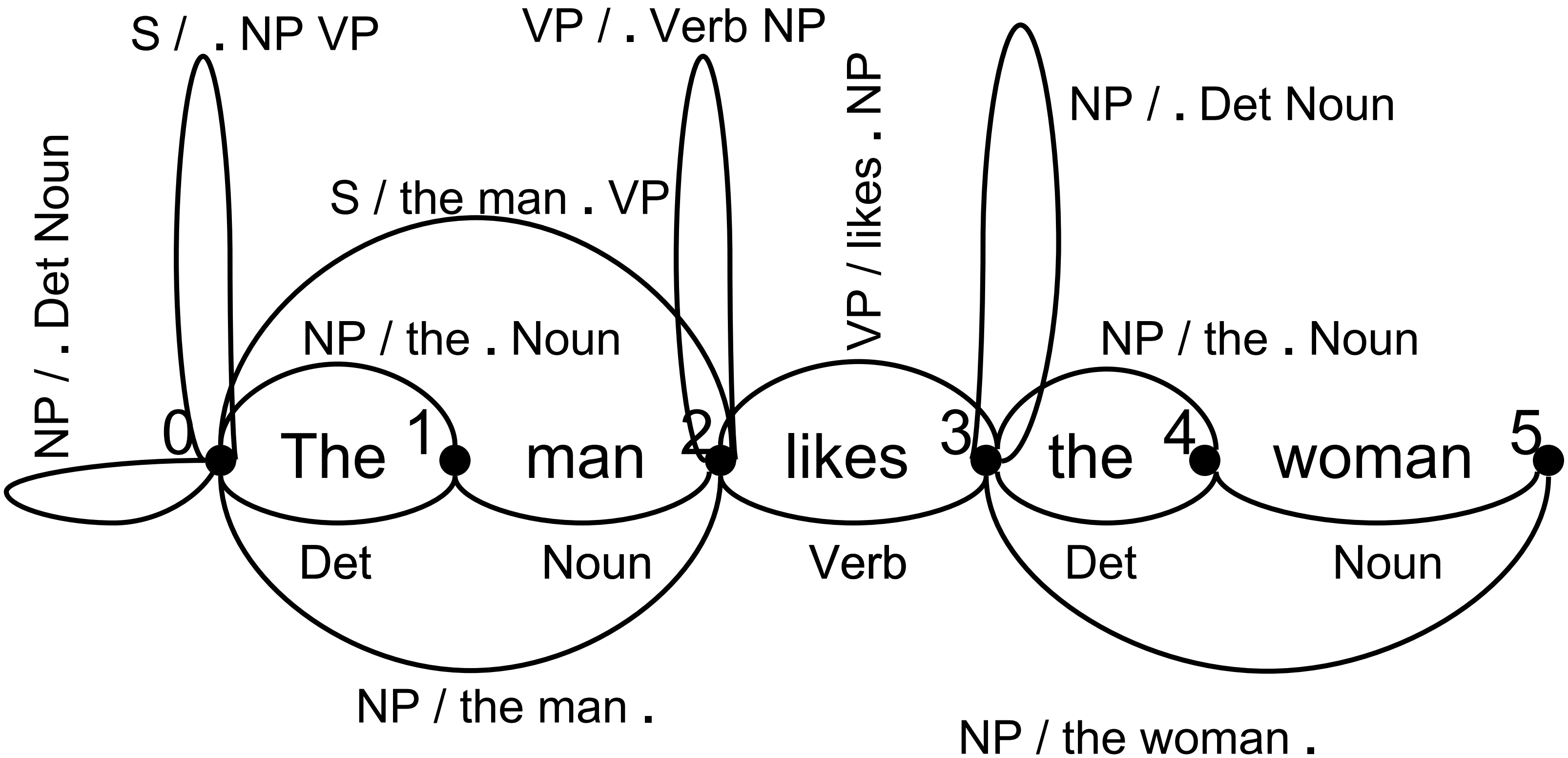
# Chart parsing - example



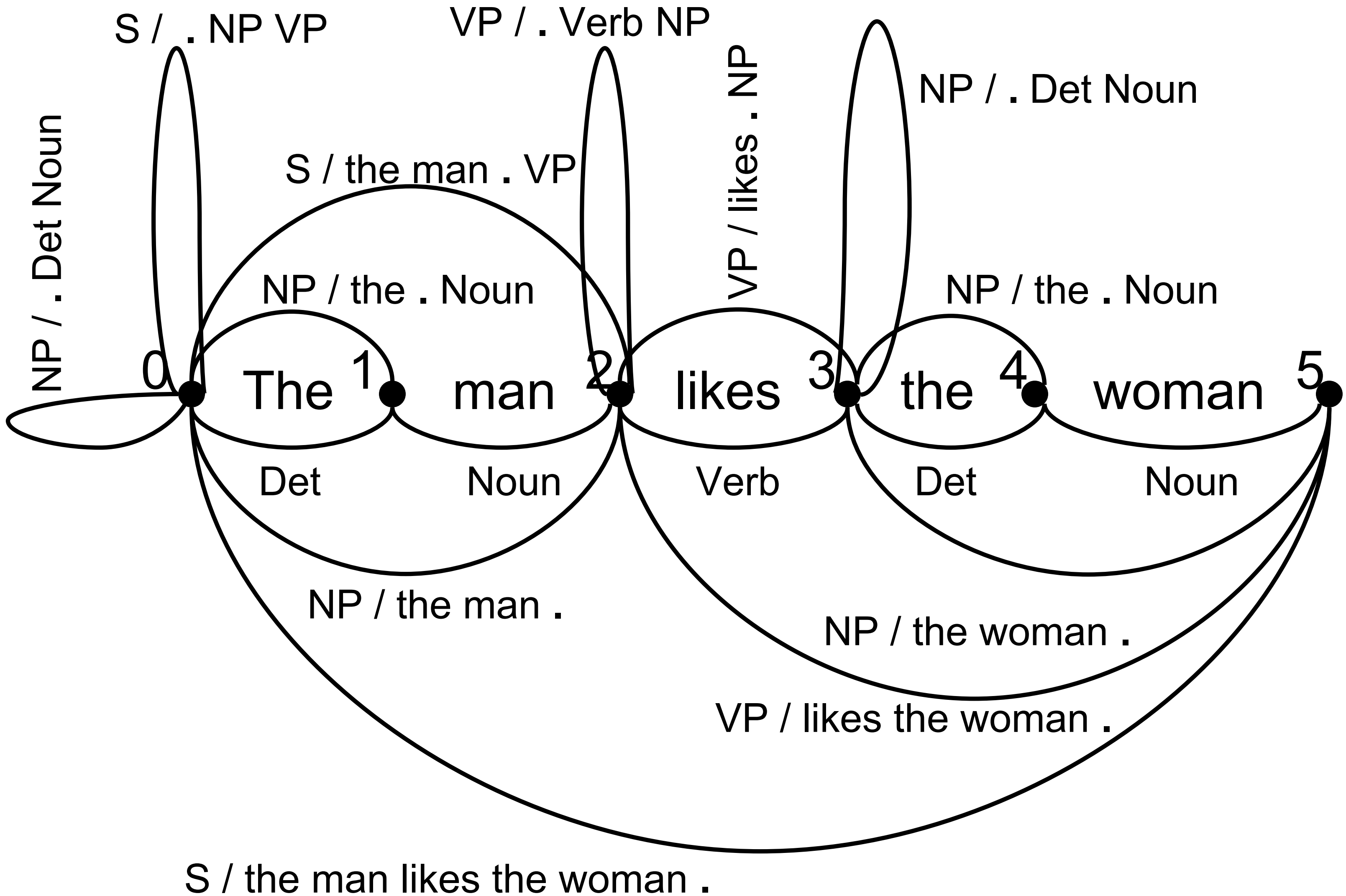
# Chart parsing - example



# Chart parsing - example



# Chart parsing - example



# Parsing algorithms - Summary

- Parsing is a crucial aspect of establishing **meaning** in language processing
- Basic **constraints**:
  - **Top-down:** grammar constrains which structures sentences can have in a language
  - **Bottom-up:** input constrains which rules from the grammar can apply
- **Ambiguity**
  - Structural and lexical ambiguities
  - Avoid reparsing by using a chart
  - Try out a real parser:

  http://www.link.cs.cmu.edu/link/submit-sentence-4.html

1. Human sentence comprehension: How to address the question of how sentences are comprehended.

2. Information sources used in sentence comprehension.

3. Modularity of information use in sentence comprehension? Probably not.

# How to uncover how the language processing mechanism works?

- Find input that the mechanism has difficulty with;
- Find input that the mechanism has little or no difficulty with.

# How to uncover how the language processing mechanism works?

Useful evidence:

1. Ambiguous input that is easy / hard to process.
2. Unambiguous input that is easy / hard to process.

# Easy to process temporary ambiguity

John knows Mary.
John knows Mary is intelligent.

The desert trains young people to be tough.
The desert trains are tough on young people.

Is the crowd in the room?
Is the crowd in the room happy?

# Hard to process temporary ambiguity: Garden-path effects

# The dog walked to the park chewed the bone.
(cf. The dog that was walked to the park chewed the bone.)

# The horse raced past the barn fell.
(cf. The horse that was raced past the barn fell.

# The cotton clothing is made of comes from Mississippi.
(cf. The cotton that clothing is made of comes from Mississippi.)

# I put the candy on the table into my mouth.
(cf. I put the candy that was on the table into my mouth.)

# Reading methods: Self-paced reading, eye-tracking

# The existence of garden-path effects provides evidence:

- That language is processed on-line, as it is heard or read
- That the human parser is not unlimited parallel.  Rather, it must be ranked parallel or serial.

# Hard to process unambiguous sentences
# Nested (or *center-embedded*) structures

The reporter disliked the editor.

The reporter [ who the senator attacked ] disliked the editor.

# The reporter [ who the senator [ who John met ] attacked ] disliked the editor.

# What sources of information do people use in processing sentences?

- Syntactic structure
- Word frequency
- Plausibility
  - (1) The dog bit the man.
  - (2) The man bit the dog.
- Discourse context
- Syntactic complexity
- Intonational information

# Information that is used in sentence comprehension

**1. Syntax: Word order**

The dog bit the boy.

vs.

The boy bit the dog.

# Information that is used in sentence comprehension

**2. Lexical (Word) information, e.g., frequency**

Unambiguous sentences: more frequent, faster: “class” vs. “caste”

Ambiguity: more frequent usages are preferred
# The old man the boats.

Syntactic argument structure frequencies
E.g., many verbs can take either an NP or a CP complement

Mary discovered / believed the answer was in the back of the book.

More difficulty in comprehending the disambiguating region “was in the ...” for the NP-biased verb “discover” than for the CP-biased verb “believe”.

# Information that is used in sentence comprehension

**2. Lexical (Word) information, e.g., frequency**

Words with multiple senses of roughly equal frequency are comprehended slower (e.g., “pitcher”) than unambiguous words or words which are highly frequency-biased towards one sense (e.g., “port”).

# Information that is used in sentence comprehension

**3. Plausibility of the resulting linguistic expression, in the world**

Unambiguous examples:
The dog bit the boy.  vs. The boy bit the dog.

Ambiguity: (Trueswell, Tanenhaus & Garnsey, 1994)
*The defendant examined* by the lawyer turned out to be unreliable.
*The evidence examined* by the lawyer turned out to be unreliable.

# Information that is used in sentence comprehension

**4. Context** (Crain & Steedman, 1985; Altmann & Steedman, 1988; Tanenhaus et al., 1995)

Ambiguity:

There were two defendants, one of whom the lawyer ignored entirely, and the other of whom the lawyer interrogated for two hours.

The defendant examined by the lawyer turned out to be unreliable.

# Monitoring visual eye-movements while listening to spoken instructions
# “Put the frog on the napkin into the box.”

Photo removed for copyright reasons.

# Monitoring visual eye-movements while listening to spoken instructions
# “Put the frog on the napkin into the box.”

Photo removed for copyright reasons.

# Monitoring visual eye-movements while listening to spoken instructions
# “Put the frog on the napkin into the box.”

Two frog context: No looks to the incorrect target (the second napkin)

Photo removed for copyright reasons.

# Monitoring visual eye-movements while listening to spoken instructions
# "Put the frog on the napkin into the box."

Two frog context: No looks to the incorrect target (the second napkin)

Photo removed for copyright reasons.

One frog context: Many looks to the incorrect target (the second napkin)